VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgaum-590018

A PROJECT REPORT **(15CSP85)** ON

“PREDICTING CYBERBULLYING IN SOCIAL MEDIA USING MACHINE LEARNING ALGORITHM”

Submitted in Partial fulfillment of the Requirements for the Degree of Bachelor of Engineering in Computer Science & Engineering

By

USMI MUKHERJEE(1CR16CS183)
VEDHITHA S (1CR16CS176)
REENA(1CR16CS130)
RIYA FRANCIS(1CR16CS134)

Under the Guidance of,

Mrs. Savitha N J

Assistant Professor, Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI,BANGALORE-560037

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CELEBRATING 25 YEARS
CMRIT
CMR INSTITUTE OF TECHNOLOGY, BENGALURU.
ACCREDITED WITH A+ GRADE BY NAAC

CERTIFICATE

Certified that the project work entitled **"Predicting Cyberbullying on Social Media using Machine Learning Algorithm"** carried out by Ms. Usmi Mukherjee, USN 1CR16CS183, Ms. Vedhitha S, USN 1CR16CS176, Ms. Reena USN 1CR16CS130, Ms. Riya Francis, USN 1CR16CS134, bonafide students of CMR Institute of Technology, in partial fulfillment for the award of **Bachelor of Engineering** in Computer Science and Engineering of the Visvesvaraya Technological University, Belgaum during the year 2019-2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

.

Mrs. Savitha N.J	**Dr. Prem Kumar Ramesh**	**Dr.Sanjay Jain**
Assistant Professor	**Professor & Head**	**Principal**
Dept. of CSE, CMRIT	**Dept. of CSE, CMRIT**	**CMRIT**

External Viva

Name of the examiners	Signature with date
1.	________________
2.	________________

DECLARATION

We, the students of Computer Science and Engineering, CMR Institute of Technology, Bangalore declare that the work entitled "Predicting Cyberbullying in Social Media using Machine Learning Algorithm" has been successfully completed under the guidance of Prof. Savitha NJ, Computer Science and Engineering Department, CMR Institute of technology, Bangalore. This dissertation work is submitted in partial fulfillment of the requirements for the award of Degree of Bachelor of Engineering in Computer Science and Engineering during the academic year 2019 - 2020. Further the matter embodied in the project report has not been submitted previously by anybody for the award of any degree or diploma to any university.

Place:
Date:

Team members:

USMI MUKHERJEE(1CR16CS183) ____________________

VEDHITHA S(1CR16CS176) ____________________

REENA(1CR16CS130) ____________________

RIYA FRANCIS (1CR16CS134) ____________________

ABSTRACT

Social technologies have created a revolution in user-generated information, online human networks, and rich human behavior-related data. Cyberbullying is a form of bullying or harassment using the electronic means. It is using technological platform to bully or attack the people using social media platform such as Twitter, Facebook, Whats App**.** There is a positive side of the social media platforms, it enables communication among people all over the world. However, the misuse of social technologies such as social media platforms, has introduced a new form of aggression and violence that occurs exclusively online. A new means of demonstrating aggressive behavior in Social Media websites are highlighted through this project. We will comprehensively review cyberbullying prediction models and identify the main issues related to the construction of cyberbullying prediction models in Social Media and build a process for cyberbullying detection and implement our model in our social media application. Data collection and feature engineering process will be done and feature selection algorithms will be applied. Finally, various machine learning algorithms will be applied for the prediction of cyberbullying behaviors. The issues and challenges will also be highlighted as well, which will present new research directions for researchers to explore.

ACKNOWLEDGEMENT

We take this opportunity to express our sincere gratitude and respect to **CMR Institute of Technology, Bengaluru** for providing us a platform to pursue our studies and carry out our final year project

We have a great pleasure in expressing our deep sense of gratitude to **Dr. Sanjay Jain,** Principal, CMRIT, Bangalore, for his constant encouragement.

We would like to thank **Dr. Prem Kumar Ramesh,** HOD, Department of Computer Science and Engineering, CMRIT, Bangalore, who has been a constant support and encouragement throughout the course of this project.

We consider it a privilege and honor to express my sincere gratitude to our guide **Prof. Savitha N J, Assistant Professor,** Department of Computer Science and Engineering, for the valuable guidance throughout the tenure of this review.

We also extend our thanks to all the faculty of Computer Science and Engineering who directly or indirectly encouraged us.

Finally, we would like to thank our parents and friends for all their moral support they have given us during the completion of this work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CNN	**Convolution Neural Network**
SM	**Social Media**

CHAPTER 1

INTRODUCTION

Cyberbullying is a major problem and has been documented as a serious national health problem due to the recent growth of online communication and social media websites. Research has shown that cyberbullying exerts negative effects on the psychological and physical health and academic performance of people. Studies have also shown that cyberbullying victims incur a high risk of suicidal ideation.

Escaping from cyberbullying is difficult because cyberbullying can reach victims anywhere and anytime. It can be committed by posting comments and statuses and pictures for a large potential audience. The victims cannot stop the spread of such activities. The nature of social media websites allows cyberbullying to occur secretly, spread rapidly, and continue easily.

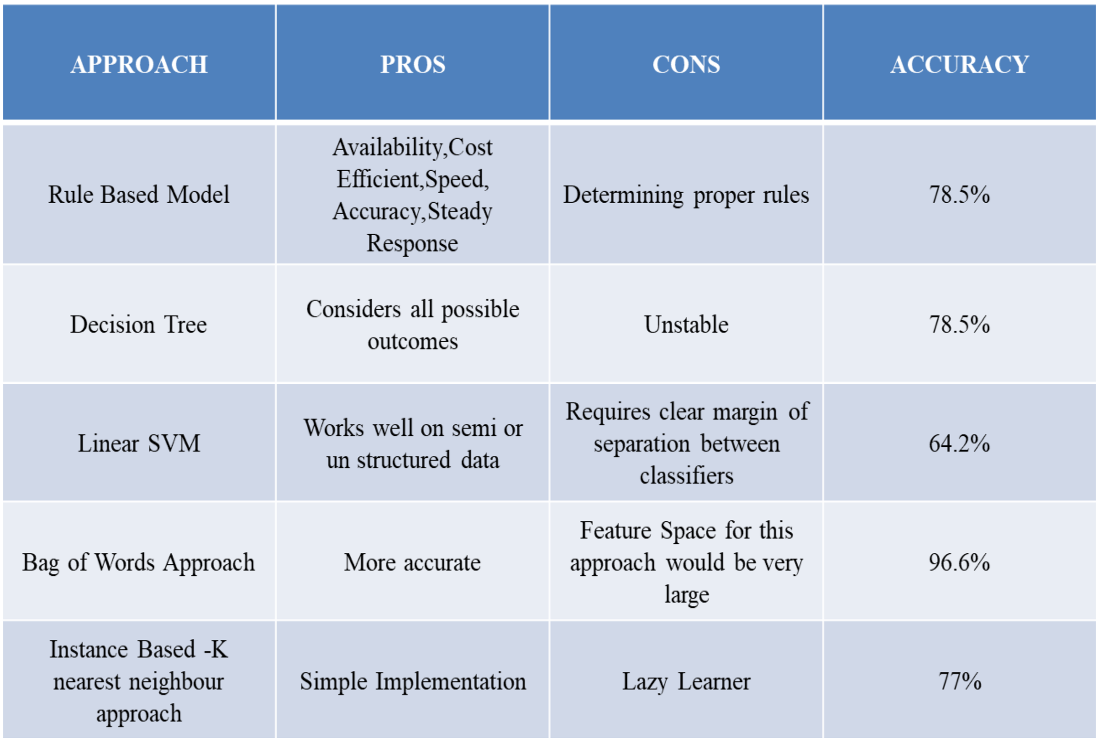
1.1 Relevance of the Project

Cyberbullying can be easily committed; it is considered a dangerous and fast-spreading aggressive behavior. Bullies only require willingness and a laptop or cell phone with Internet connection to perform misbehavior irrespective of geographical location and hence without confronting victims. Developing an effective prediction model for predicting cyberbullying is therefore of practical significance. Developing a cyberbullying prediction model that detects aggressive behavior that is related to the security of human beings is more important than developing a prediction model for aggressive behavior related to the security of machines

1.1.1 Summary of the approaches

The best approach we found is sentimental analysis. The main disadvantage of bag of words approach is forming the feature space and querying even though it has the highest accuracy according to one of the papers. Rule Based Model is used to determine proper rules. Linear SVM requires clear margin of separation between classifiers. Bag of Words Approach the feature of space is large.

Table 1.1 Summary of Approaches



APPROACH	PROS	CONS	ACCURACY
Rule Based Model	Availability,Cost Efficient,Speed, Accuracy,Steady Response	Determining proper rules	78.5%
Decision Tree	Considers all possible outcomes	Unstable	78.5%
Linear SVM	Works well on semi or un structured data	Requires clear margin of separation between classifiers	64.2%
Bag of Words Approach	More accurate	Feature Space for this approach would be very large	96.6%
Instance Based -K nearest neighbour approach	Simple Implementation	Lazy Learner	77%

1.2 Scope of the Project

Cyberbullying is a major problem and has been documented as a serious national health problem due to the recent growth of online communication and Social Media websites. Research has shown that cyberbullying exerts negative effects on the psychological and physical health and academic performance of people. Cyberbullying victims incur a high risk of suicidal ideation. Consequently, developing a cyberbullying prediction model that detects aggressive behavior that is related to the security of human beings is more important than developing a prediction model for aggressive behavior related to the security of machines. Social Media websites have become an integral part of user's lives; a study found that Social Media websites are the most common platforms for cyberbullying victimization. Detection of such cyberbullying attacks becomes absolutely necessary.

1.3 Problem Statement

Detecting cyberbullying messages by data collection, feature engineering, constructing cyberbullying detection model using the Convolutional neural network (CNN) approach is suitable for our model. CNN has high statistical and computational efficiency and can handle massive amount of data. Implementing them in our social media application. Removing the bullying comments from the application upon prediction.

1.4 Agile Methodology

After reviewing the different classifiers. The model uses Convolutional Neural Network approach as it can handle massive amount of data and prevent data sparsity. The labelled dataset is fetched from the Kaggle and Super data science websites. After that the cleaning of the data is performed. Cleaning refers to word correction and removing of unwanted symbols, URL links, white spaces. The dataset is then loaded. The Convolutional Neural Network model is built and the hyperparameters are set for the model. The detecting model undergoes phases of training and testing. Implementation of the prediction model is done using Django application. And if the user posts a comment, the prediction model is run on the comment in order to determine if posted comment is bullying comment or not.

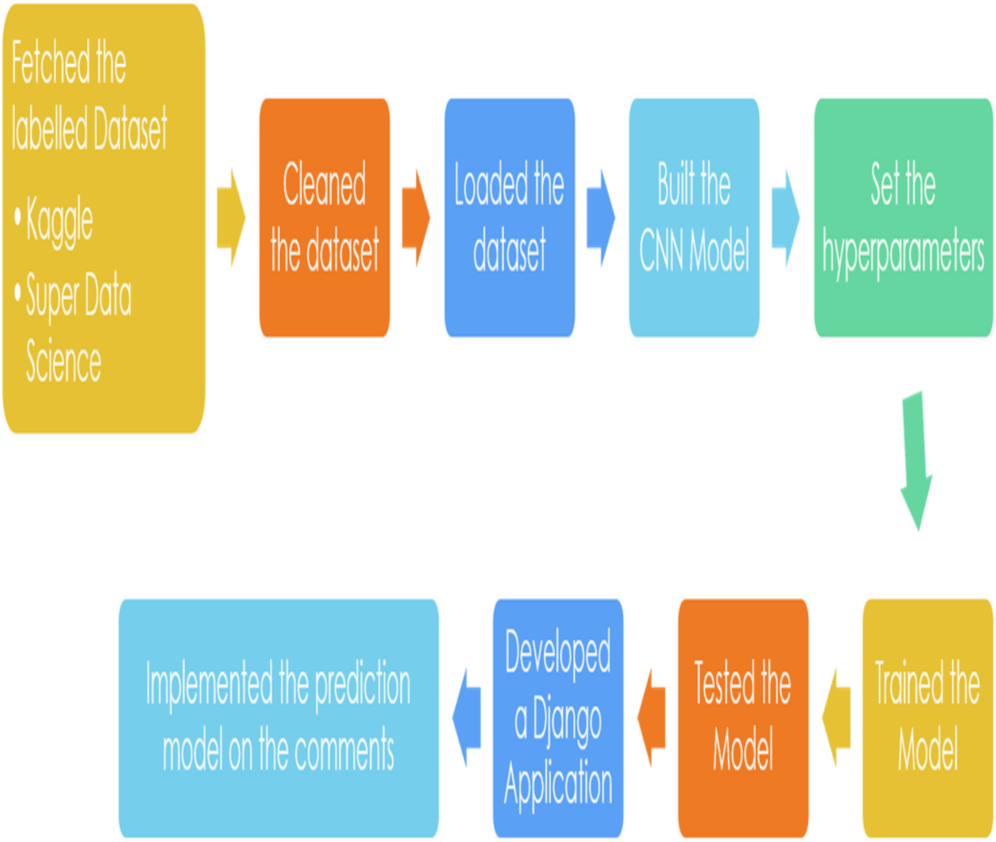


Fig 1.4 Agile Methodology

1.5 Proposed System

The proposed system consists of the following features:

• The designed application is a blog. The user has to register himself in the blog.

• The user has authority to post messages and comment on the different posts also.

• He/She can send as many as comments they wish on the particular post without even logging in the blog.

• If the user's comment is of a bullying content. The predictive model runs on the comments in order to determine the type of texts and classifies them as bullying and non-bullying.

• If the comment is bullying, the particular comment is removed and prevents the user to put such bullying comments.

• Since the model uses Convolutional Neural Network the accuracy in detecting the bullying messages is high compared to the other classifiers such as the SVM and the Naïve Bayes.

1.7 Chapter wise summary

Here is a chapter wise summary of this report.

• Chapter 2: Literature Survey

Wereviewed3researchpapersandcomparedthemtofindthebestapproachfor doing the project. This helps us to determine the most suitable approach for the model.

• Chapter 3: System Requirements Specification

This chapter contains all the hardware and software requirements.

• Chapter 4: System Analysis and Design

In this chapter we described the approach for preparing our model i.e. the Convolutional Neural Network.

• Chapter 5: Implementation

This chapter contains how the detailed implementation and phases.

• Chapter 6: Results and Discussion

This chapter contains the results we observed and obtained from the project

- **Chapter 7: Testing**

  This chapter contains the various testing methods used to test the project.

- **Chapter 8: Conclusion**

  This chapter contains the various conclusions drawn from our work

CHAPTER 2

LITERATURE SURVEY

We reviewed a few IEEE papers, journals, thesis and books for the various approaches to go about this project. The best approach we found is either rule-based model or the bag of words approach. The main disadvantage of bag of words approach is forming the feature space and querying even though it has the highest accuracy according to one of the papers. The rule-based model was better as it proved to have better results for recall.

2.1 Approaches to automated detection of Cyberbullying: A Survey

2.1.1 Abstract:

Research into cyberbullying detection has increased in recent years, due in part to the proliferation of cyberbullying across social media and its detrimental effect on young people. A growing body of work is emerging on automated approaches to cyberbullying detection. These approaches utilize machine learning and natural language processing techniques to identify the characteristics of a cyberbullying exchange and automatically detect cyberbullying by matching textual data to the identified traits. In [1] presents a systematic review of published research on cyberbullying detection approaches. On the basis of our extensive researches being carried out, the existing approaches are categorized into 4 main classes, namely supervised learning, lexicon-based, rule-based, and mixed-initiative approaches. We found lack of labelled datasets and non-holistic consideration of cyberbullying by researchers when developing detection systems are two key challenges facing cyberbullying detection research.[1] essentially maps out the state-of-the- art in cyberbullying detection research and serves as a resource for researchers to determine where to best direct their future research efforts in this field.

2.1.2 Approach:

1. Supervised learning-based approaches typically use classifiers such as SVM and Naïve Bayes to develop predictive models for cyberbullying detection.

2. Lexicon-based systems utilize word lists and use the presence of words within the lists to detect cyberbullying.

3. Rule-based approaches match text to predefined rules to identify bullying,

4. Mixed-initiatives approaches combine human-based reasoning with one or more of the mentioned approaches.

2.1.3 Other Approaches:

Aside from supervised (and semi-supervised) learning, mixed-initiative, rule-based, and lexicon-based cyberbullying detection systems, we found a number of papers that employ approaches that do not easily lend themselves to our categorizations. Such papers include Bosse and Stam [1]in the year 2011 where the authors formulated the detection problem as a norm violation issue by introducing a number of normative agents into a virtual environment to monitor the activities of users within the virtual world. Mancilla- Caceresetal.alsostudieduserinteractionswithinavirtualenvironment. They created a social computer game that required players to create teams and work collaboratively together to perform tasks. Using 5th grade students as case studies, they observed the student's behaviors within the game and compared this to the results of a survey administered by cyberbullying experts to the same group of students prior to the game. By analyzing interactions within the game, they discovered a collective attempt by a number of students to bully another student.

2.1.4 Conclusion:

The current state of affairs for cyberbullying prevention within online social networks therefore requires urgent attention and improvement. This improvement is only possible if the research community, educational institutions, law enforcement, social media platforms, and software vendors make conscious and concerted efforts to facilitate the diffusion of knowledge and expertise in all directions. It is only when this happens that viable cyberbullying detection applications can advance beyond research boundaries into the wider world. In [1] they used **Naïve Baye**s for the prediction model. In Naïve Bayes if a categorical variable has a category in the test dataset.

Not observed in training dataset, then the model will assign a 0 (zero) probability and will be unable to make a prediction

2.2Online Social Network Bullying Detection Using Intelligence Techniques

2.2.1 Abstract:

Social networking sites (SNS) is being rapidly increased in recent years, which provides platform to connect people all over the world and share their interests. However, the Social Networking Sites is providing opportunities for cyberbullying activities. Cyberbullying is harassing or insulting a person by sending messages of hurting or threatening nature using electronic communication. Cyberbullying poses significant threat to physical and mental health of the victims.

In [2] detection of cyberbullying and the provision of subsequent preventive measures are the main courses of action to combat cyberbullying. The proposed method is an effective method to detect cyberbullying activities on social media. The detection method can identify the presence of cyberbullying terms and classify cyberbullying activities in social network such as Flaming, Harassment, Racism and Terrorism, using Fuzzy logic and Genetic algorithm. The effectiveness of the system is increased using Fuzzy rule set to retrieve relevant data for classification from the input. In the proposed method Genetic algorithm is also used, for optimizing the parameters and to obtain precise output.

2.2.2 Approach:

- In the proposed architecture the process of detecting cyberbully activities begins with input dataset from social network. Input is text conversation collected fromformspring.me.

- Input is given to data pre-processing which is applied to improve the quality of the research data and subsequent analytical steps, this includes removing stop words, extra characters and hyperlinks. This step helps to clean the dataset and remove unwanted symbols and retain only the texts necessary to be processed.

- After performing pre-processing on the input data, it is given to Feature Extraction. Feature Extraction is done to obtain features like Noun, Adjective and Pronoun from the text and statistics on occurrence of word (frequency) in the text.

- The extracted features are given to Learning Algorithm. The Learning algorithm unit is the central element of the architecture and is composed of a genetic algorithm for modeling adaptive and exploratory behavior. Knowledge is given as Fuzzy rule set. The main functionality is to adjust the representation of the information needed for classification and yet retains the essential knowledge from the past. This knowledge is kept in a population of chromosomes, which is processed by the genetic algorithm.

- All the chromosomes in the population are competing to predict the classification of cyberbully activities.

- The output from learning unit is given to Classifier technique classifies the cyberbully activities using the fitness value of chromosome. The ability of a chromosome to classify the activity is called the fitness of the chromosome.

- The chromosome with higher fitness value gives the classified output. The output is classified bullying words present in the conversation.

2.2.3 Conclusion:

The proposed system focuses on detecting the presence of cyberbullying activity in social networks using fuzzy logic which helps government to take action before many users become victim of cyberbullying. The system also uses genetic operators like crossover and mutation for optimizing the parameters and obtain precise type of cyberbullying activity which helps government or other social welfare organization to identify the cyberbullying activities in social network and to classify it as Flaming, Harassment, Racism or Terrorism and take necessary actions to prevent the users of the social network from becoming victims. In [2] **Genetic algorithm** is used as a learning algorithm; Knowledge is given as a fuzzy rule set and general classifier is used to classify the output based on the fitness value. It isn’t a suitable method as it may not find the optimal solution to the problem in all cases.

2.2.4 Output based on the approach

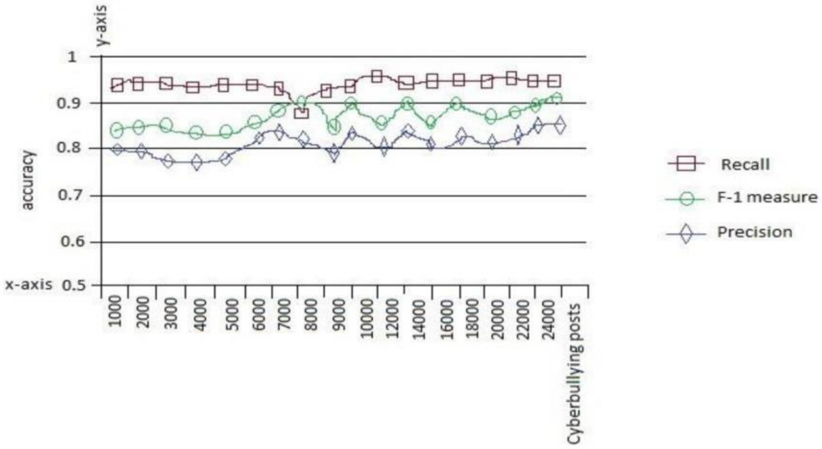


Fig 2.2 Research paper 2 Output

2.3 Automatic detection of cyberbullying in social media text

2.3.1 Abstract:

While social media offer great communication opportunities, they also increase the vulnerability of young people to threatening situations online. Recent studies report that cyberbullying constitutes a growing problem among youngsters. Successful prevention depends on the adequate detection of potentially harmful messages and the information overload on the Web requires intelligent systems to identify potential risks automatically. The focus of [3] is on automatic cyberbullying detection in social media text by modelling posts written by bullies, victims, and bystanders of online bullying. [3] describes the collection and fine-grained annotation of a cyberbullying corpus for English and Dutch and perform a series of binary classification experiments to determine the feasibility of automatic cyberbullying detection. We make use of linear support vector machines exploiting a rich feature set and investigate which information sources contribute the most for the task. Experiments on a hold-out test set reveal promising results for the detection of cyberbullying-related posts. After optimization of the hyper-parameters, the classifier yields an F_1 score of 64% and 61% for English and Dutch respectively, and considerably outperforms baseline systems. In [3] the proposed model uses Support Vector Machine as its classifier.

2.3.2 Approach:

They proposed a method based on a linear SVM classifier exploiting a rich feature set. The contribution they make is twofold: first, they develop a complex classifier to detect signals of cyberbullying, which allows them to detect different types of cyberbullying that are related to different social roles involved in a cyberbullying event. Second, they demonstrate that the methodology is easily portable to other languages.

2.3.3 Steps:

- The first step is the construction of two corpora, English and Dutch, containing social media posts that are manually annotated for cyberbullying according to our fine-grained annotation
- Two corpora were constructed by collecting data from the social networking site.
- Then the corpora are annotated using brat rapid annotation tool
- Firstly, the annotators were asked to indicate, at the message or post level, whether the text under investigation was related to cyberbullying. If the message was considered harmful and thus contained indications of cyberbullying, annotators identified the author's participant role. Message level

  - Harraser/bully, victim, bystander-defender, bystander- assistant
- Secondly, at the sub-sentence level, the annotators were tasked with the identification of fine-grained text categories related to cyberbullying. Sub sentence Level-Threat, Blackmail, Insult, Curse, etc
- Annotator statistics are calculated using inter annotator agreement scores were calculated.
- A cost sensitive SVM was applied on it as a hyperparameter
- Binary classification experiments using a linear kernel support vector machine (SVM) implemented in LIBLINEAR by making use of Scikit-learn

The classifier was optimized for feature type and hyper-parameter combinations. Model selection was done using 10-fold cross validation in grid search over all possible feature types (i.e. groups of similar features, like different orders of n-gram bag-of-words features) and hyper-parameter configurations.

- The best performing hyper-parameters are selected by F1 score on the positive class. The winning model is then retrained on all held-in data and subsequently tested on a hold-out test set to assess whether the classifier is over- or under- fitting. F1 score is s statistical analysis of binary classification, the score is measure of a test's accuracy. It considers both the precision *p* and the recall r of the test to compute the score: p is the number of correct positive results divided by the number of all positive results returned by the classifier, and r is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive).

- The optimized models are evaluated against two baseline systems: i) an unoptimized linear-kernel SVM (configured with default parameter settings) based on word n-grams only and Linear Kernel is used when the data is Linearly separable, that is, it can be separated using a single Line. It is one of the most common kernels to be used. It is mostly used when there are a Large number of Features in a particular Data Set. One of the examples where there are a lot of features, is Text Classification, as each alphabet is a new feature. So we mostly use Linear Kernel in Text Classification., ii) a keyword-based system that marks posts as positive for cyberbullying if they contain a word from existing vocabulary lists composed by aggressive language and profanity terms.

- Person alternation is a binary feature indicating whether the combination of a first- and second-person pronoun occurs in order to capture interpersonal intent.

- Subjectivity lexicon features defines positive and negative opinion word ratios, as well as the overall post polarity were calculated using existing sentiment lexicons. For Dutch, we made use of the Duoman and Pattern lexicons. For English, we included the Liu and Hu opinion lexicon, the MPQA lexicon, the General Inquirer Sentiment Lexicon .

3.3.4 Conclusion:

The main contribution of this paper is that it presents a system to automatically detect signals of cyberbullying on social media, including different types of cyberbullying, covering posts from bullies, victims and bystanders. We evaluated our system on a manually annotated cyberbullying corpus for English and Dutch and hereby demonstrated that our approach can easily be applied to different languages, provided that annotated data for these languages are available. In [3] **SVM** is used for the prediction model It is not suitable for large datasets. SVM does not perform very well, when the dataset has more noise i.e. target classes are overlapping.

CHAPTER 3

REQUIREMENTS SPECIFICATIONS

Requirement Specification detailed description of a software system to be developed with its functional and non-functional requirements. It may include the use cases of how user is going to interact with software system and the requirements of the system for execution of the project . The requirement specification consistent of all necessary requirements required for project development. To develop the software system we should have clear understanding of Software system. To achieve this we need to continuous communication with customers to gather all requirements.

3.1 Hardware Requirements

• Processor-Intel or AMD(Advanced Micro Devices)

• RAM-8GB(minimum)

• HardDisk-1TB(minimum)

• Mouse

• Keyboard

• Monitor

3.2 Software Requirements

The system contains software:

- Text Editor- visual studio code or any other
- Python 3.0 or above
- Jupyter Notebook
- Python and machine learning libraries
- Django
- PostgreSQL

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

We will be using convolutional neural network (CNN or Conv Net) for construction of the cyberbullying prediction model.

4.1 Problem Statement

Machine learning algorithms provide an opportunity to effectively predict and detect negative forms of human behavior, such as cyberbullying. Developing an effective prediction model for predicting cyberbullying is therefore of practical significance.

Detecting cyberbully in messages by using machine learning approaches namely, data collection, feature engineering, construction of cyberbullying detection model, and evaluation of constructed cyberbullying detection models and also implementing them in our social media application.

4.2 Convolutional neural network

4.2.1 Background

Here every neuron in a layer is being connected to all neurons in the next layer and is hence referred to as fully connected networks.

Convolutional Neural Networks have the following layers:

Convolution Layer

It is the input layer. This layer is the basic unit of a CNN. The user-specified parameters enter this network through this layer. A convolutional layer should have the following:

- width and height are used to define this layer
- number of entry and exit channels

This takes the input and sends the result to the next layer which is similar to the working of a neuron. Every convolutional neuron acts on the data for its respective portion of the sensory space.

ReLU Layer

The most popularly used activation function in neural networks is ReLU layer , which is used mainly in CNN. It applies the non-saturating activation function . It removes non- positive values by converting them to zero values .The nonlinear properties of the decision function and of the overall network are increased by ReLU . ReLU is quick to evaluate. Introducing nonlinearity to a system that has just been computing several linear operations during the previous convolution layers is the main purpose of this layer . ReLU layers work far better and trains a lot faster without causing much change in the accuracy values.

Pooling Layer

Local or global pooling layers are included in Convolutional networks. By combining the outputs of neuron clusters at one layer into a single neuron in the next layer they reduce dimensions of the data Small clusters are combined in Local Pooling. All the neurons of the convolutional layer are acted upon in the Global Pooling step. The maximum value from each of a cluster of neurons at prior layers are used in Max pooling Average values from each of a cluster of neurons are used in Average Pooling in prior layers. For implementing pooling any non-linear functions are used and max pooling is the most popular. Reduction of the spatial size of the representation, reduction of the number of parameters, memory footprint and amount of computation in the network, and for control overfitting pooling layers can be used

Fully connected Layer

In fully connected layers , every neuron in a layer is connected to every other neuron in another layer. High-level reasoning is done after several convolutional and max pooling layers, through these fully connected layers. Like in regular (non-convolutional) artificial neural networks , neurons in this layer have connections to all activations in the previous layers.

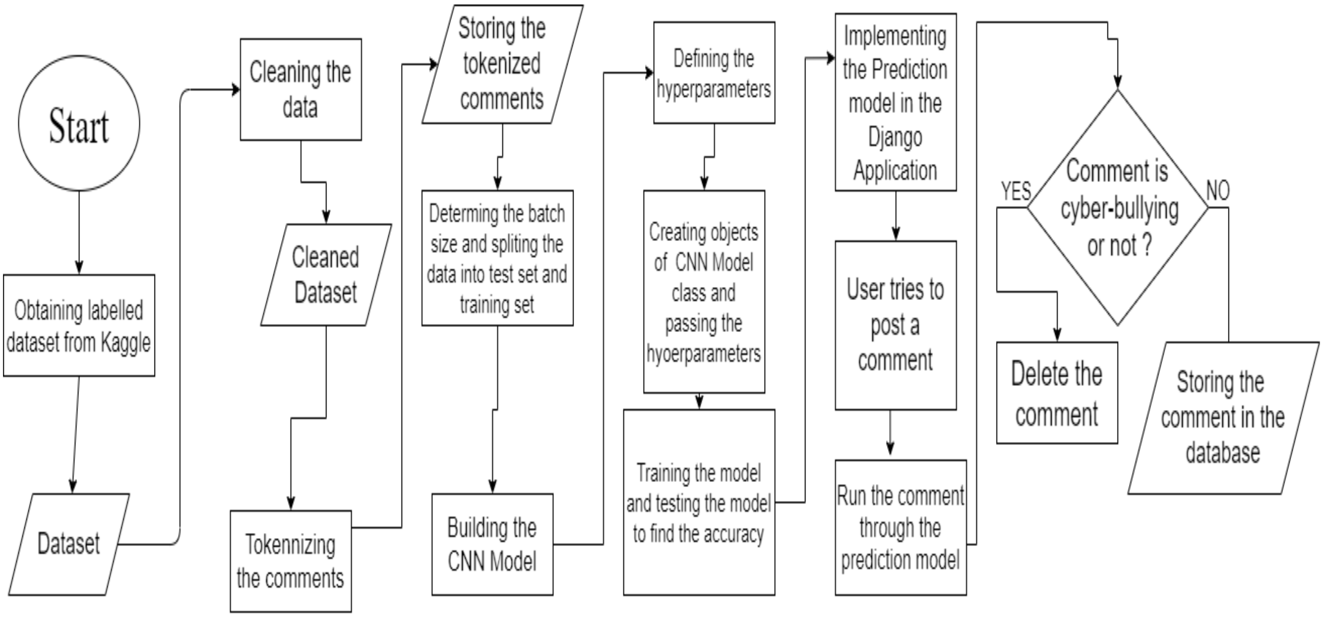

Start
Obtaining labelled dataset from Kaggle
Dataset
Cleaning the data
Cleaned Dataset
Tokennizing the comments
Storing the tokenized comments
Determing the batch size and spliting the data into test set and training set
Building the CNN Model
Defining the hyperparameters
Creating objects of CNN Model class and passing the hyoerparameters
Training the model and testing the model to find the accuracy
Implementing the Prediction model in the Django Application
User tries to post a comment
Run the comment through the prediction model
Comment is cyber-bullying or not ?
YES
NO
Delete the comment
Storing the comment in the database

Fig 4.2 CNN procedure

4.2.2 Experimental Setup

- We will be obtaining the labelled dataset from Kaggle.
- The datasets need to be cleaned which will be done by removing the stop words and whitespaces. This information is not important to the classifier.
- We will be performing tokenization. Here the cleaned dataset will be taken as sentences or paragraphs and output it as separate words.
- After this we will be determining the batch size and splitting the data into test set and training set.
- We will be building the CNN model and defining the hyperparameters.
- We will be creating the objects of the CNN model class and passing the hyperparameters.
- We will be training the model and testing the model to find the accuracy.
- We will be implementing the prediction model in the Django application.
- When the user tries to post a comment, the comment is run through the prediction model
- The comment is checked if it is cyberbullying one or not.
- If it is a cyberbullying comment , the comment is deleted else the comment is stored in the database.

CHAPTER 5

IMPLEMENTATION

The algorithm used is Convolutional Neural Network. CNN classifies a neural network consists of units(neurons), arranged in layers, which convert an input vector in to some output. Each unit takes an input, applies a (often nonlinear) function to it and then passes the output on to the next layer. CNN’s are the most mature form of deep neural networks to produce the most accurate. The size of dataset used is : 64141. The total number of layers is 4. We have used 90 percent of train data and 10 percent of test data.

5.1 Code

5.1.1 Importing all the libraries and packages

```
import pandas as pd
import numpy as np
import matplotlib. pyplot as plt
import seaborn as sns
sns.set_style('whitegrid')
%matplotlib inline
import re
from sklearn import metrics
from collections import Counter
import itertools
import numpy as np
import math
import re
import pandas as pd
from bs4 import BeautifulSoup
import random
try:
```

```
except Exception:

pass

import tensorflow as tf

import tensorflow_hub ashub

from tensorflow. keras import layers

import bert
```

5.1.2 Importing and Pre-processing the dataset

Loading the dataset

```
data = pd. DataFrame ()

data = pd. read_csv ("new_dataset.csv", encoding="utf-8")

data. head ()
```

5.1.3 Plotting the labelled dataset

The labelled dataset is plotted as categories i.e. no. of cyberbullying comments (0) and no. of non-cyberbullying comments (1) versus the total number of comments.

```
y = data['label']

classes = data['label']. unique ()

plt. figure (figsize= (10,6))

sns.  countplot  (y,  order=classes,  palette='Set1')

plt. title ('Cyberbullying Detection', fontsize=20)

plt. xlabel ('Category', fontsize=16)

plt. ylabel ('Number of Cases', fontsize=16)
```

5.1.4 Function to clean every comment in the dataset

BeautifulSouplxml-parser used for parsing every comment in the dataset. Then using regular expression @, URL and whitespaces are removed. Only the letters are kept.

```
def clean_ comment(comment):

comment = BeautifulSoup (comment, "lxml"). get_text ()

# Removing the @

comment = re.sub(r"@[A-Za-z0-9]+", ' ', comment)

# Removing the URL links

comment = re.sub(r"https?://[A-Za-z0-9./]+", ' ', comment)
```

```
# Keeping only letters

comment = re.sub(r"[^a-zA-Z.!?']", ' ', comment)

# Removing additional whitespaces

comment = re.sub (r" +", ' ', comment)

return comment
```

5.1.5 Cleaning every comment in the dataset

The previously defined function for cleaning the comment is called. The cleaned comment is stored indata_clean and the label values are stored in data_labels.

```
data_clean = [clean_comment(comment) for comment in data. comment]

data_labels = data. label. Values
```

5.1.6 Creating a BERT Tokenizer

In order to use BERT text embeddings as input to train text classification model, we need to tokenize our text.Tokenization refers to dividing a sentence into individual words. To tokenize our text, we will be using the BERTtokenizer.

```
FullTokenizer = bert. bert_tokenization.FullTokenizer

bert_layer = hub.KerasLayer("https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/

trainable=False)

vocab_file = bert_layer. resolved_object.vocab_file.asset_path.numpy()

do_lower_case = bert_layer.resolved_object.do_lower_case.numpy()

tokenizer = FullTokenizer(vocab_file, do_lower_case)
```

5.1.7 Tokenizing the comments in thedataset

```
def encode_sentence(sent):

return tokenizer.convert_tokens_to_ids(tokenizer.tokenize(sent))

data_inputs = [encode_sentence(sentence) for sentence in data_clean]
```

5.1.8 Sorting Data and Converting to Tensor Flow format

•The following script creates a list of lists where each sub list contains tokenized comment, the label of the comment and the length of the comment.

•The list is then shuffled so that in the training batches both bullying and non-bullying data is present.

•Once the data is shuffled, we will sort the data by the length of the comment.

- To do so, we will use the sort () function of the list and will tell it that we want to sort the list with respect to the third item in the sub list i.e. the length of the comment.

- Once the comments are sorted by length, we can remove the length attribute from all the comments making sure the comment at least has a length of 7 (tokens for each comment)

- ThenwewillconvertthesorteddatasetintoaTensorFlow2.0-compliantinputdataset shape.

```
data_with_len = [[sent, data_labels[i], len(sent)]

for i, sent in enumerate(data_inputs)]

random.shuffle(data_with_len)

data_with_len.sort(key=lambda x: x[2])

sorted_all = [(sent_lab[0], sent_lab[1])

for sent_lab in data_with_len if sent_lab[2] > 7]
```

5.1.9 Creating the model class

- To do so, we will create a class named DCNN that inherits from the tf.keras.Model class. Inside the class we will define our model layers.

- We design 4 layers of CNN

- In the constructor of the class, we initialize some attributes with default values. These values will be replaced later on by the values passed when the object of the DCNN class is created.

- Next, three convolutional neural network layers have been initialized with the kernel orfiltervaluesof2, 3, 4 and 5 respectively. Again, you can change the filter sizes if you want.

- Next, inside the call () function, global max pooling is applied to the output of each of the convolutional neural network layer.

- Finally, the three convolutional neural network layers are concatenated together and their output is fed to the first densely connected neural network.

- The second densely connected neural network is used to predict the output sentiment since it only contains2classes.

```
class DCNN(tf.keras.Model):

definit(self, vocab_size,

emb_dim=128,

nb_filters=50,

FFN_units=512,
nb_classes=2,
dropout_rate=0.2,
training=False,
name="dcnn"):
super(DCNN, self).init(name=name)
self.embedding = layers.Embedding(vocab_size,
emb_dim)
self.bigram = layers.Conv1D(filters=nb_filters,
kernel_size=2,
padding="valid",
activation="relu")
self.trigram = layers.Conv1D(filters=nb_filters,
kernel_size=3,
padding="valid",
activation="relu")
self.fourgram = layers.Conv1D(filters=nb_filters,
kernel_size=4,
padding="valid",
activation="relu")
self.fivegram = layers.Conv1D(filters=nb_filters,
kernel_size=5,
padding="valid",
activation="relu")
self.sixgram = layers.Conv1D(filters=nb_filters,
kernel_size=6,
padding="valid",
activation="relu")
self.pool = layers.GlobalMaxPool1D()
self.dense_1 = layers.Dense(units=FFN_units, activation="relu")
self.dropout = layers.Dropout(rate=dropout_rate)
```

```
if nb_classes == 2:
self.last_dense = layers.Dense(units=1,
activation="sigmoid")
else:
self.last_dense = layers.Dense(units=nb_classes,
activation="softmax")
```

CHAPTER 6

RESULTS AND DISCUSSION

6.1 Results

We apply convolutional neural network (CNN) as this among the best performance classifiers. It is found that the accuracy of models is 91%. As a further step we add more layers to our model to achieve more accuracy. The proposed algorithm CNN-advances the current state of cyberbullying detection by providing better predictions (higher accuracy) although it eliminates the need for feature engineering.

Classifier	2-gram	3-gram	4-gram	Average
SVM	89.42%	89.90%	90.30%	89.87%
CNN	90.90%	92.80%	91.60%	91.76%

Table 6.1: Datasets Distributions

After pre-processing the datasets, we follow the same steps to extract the features. We then split the datasets in ratios for test and train Then split the dataset into ratios for train and test. Accuracy, recall and precision, and f-score are taken as a performance measure to evaluate the classifiers. Then apply Convolution Neural Network (CNN) as they are among the best performance classifiers.

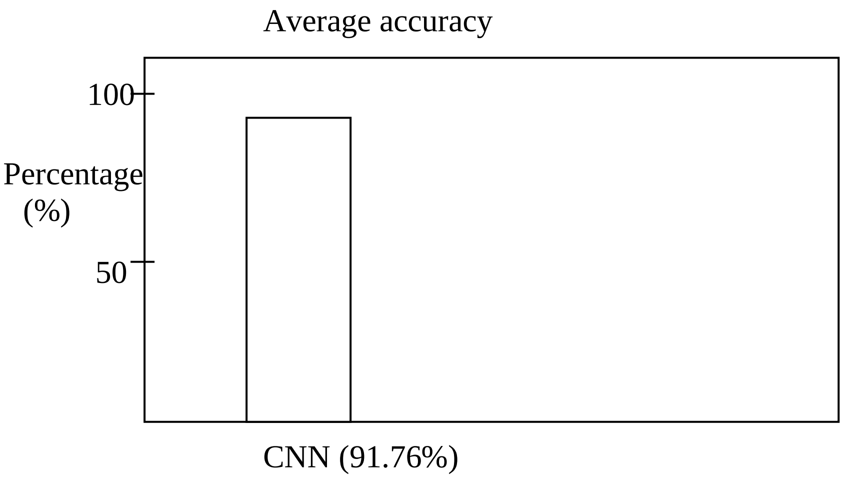
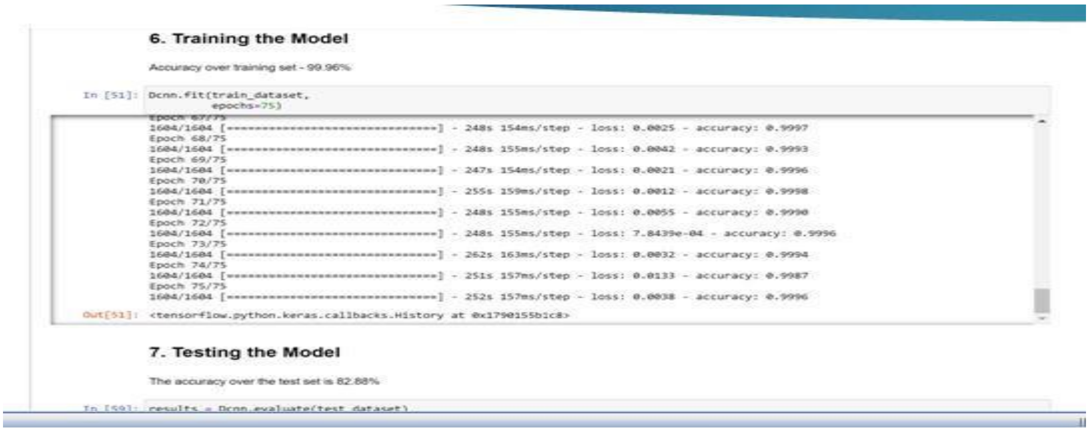


Fig. 6.1 Average Accuracy

6.2 Screenshots



6. Training the Model

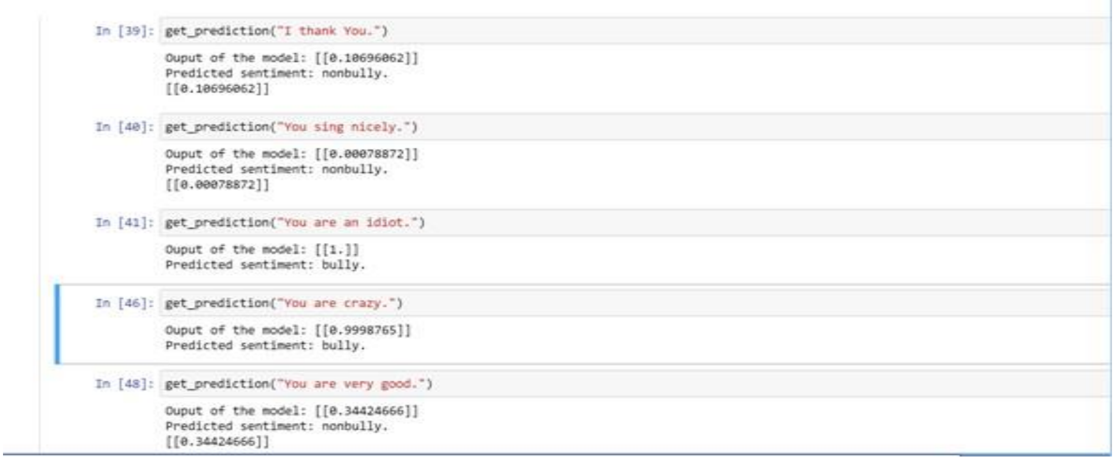
Accuracy over training set - 99.96%

```
In [51]: Dcnn.fit(train_dataset,
                  epochs=75)
Epoch 67/75
1604/1604 [==============================] - 248s 154ms/step - loss: 0.0025 - accuracy: 0.9997
Epoch 68/75
1604/1604 [==============================] - 248s 155ms/step - loss: 0.0042 - accuracy: 0.9993
Epoch 69/75
1604/1604 [==============================] - 247s 154ms/step - loss: 0.0021 - accuracy: 0.9996
Epoch 70/75
1604/1604 [==============================] - 255s 159ms/step - loss: 0.0012 - accuracy: 0.9998
Epoch 71/75
1604/1604 [==============================] - 248s 155ms/step - loss: 0.0055 - accuracy: 0.9990
Epoch 72/75
1604/1604 [==============================] - 248s 155ms/step - loss: 7.8439e-04 - accuracy: 0.9996
Epoch 73/75
1604/1604 [==============================] - 262s 163ms/step - loss: 0.0032 - accuracy: 0.9994
Epoch 74/75
1604/1604 [==============================] - 251s 157ms/step - loss: 0.0133 - accuracy: 0.9987
Epoch 75/75
1604/1604 [==============================] - 252s 157ms/step - loss: 0.0038 - accuracy: 0.9996
Out[51]: <tensorflow.python.keras.callbacks.History at 0x1790155b1c8>
```

7. Testing the Model

The accuracy over the test set is 82.88%

```
In [59]: results = Dcnn.evaluate(test_dataset)
```

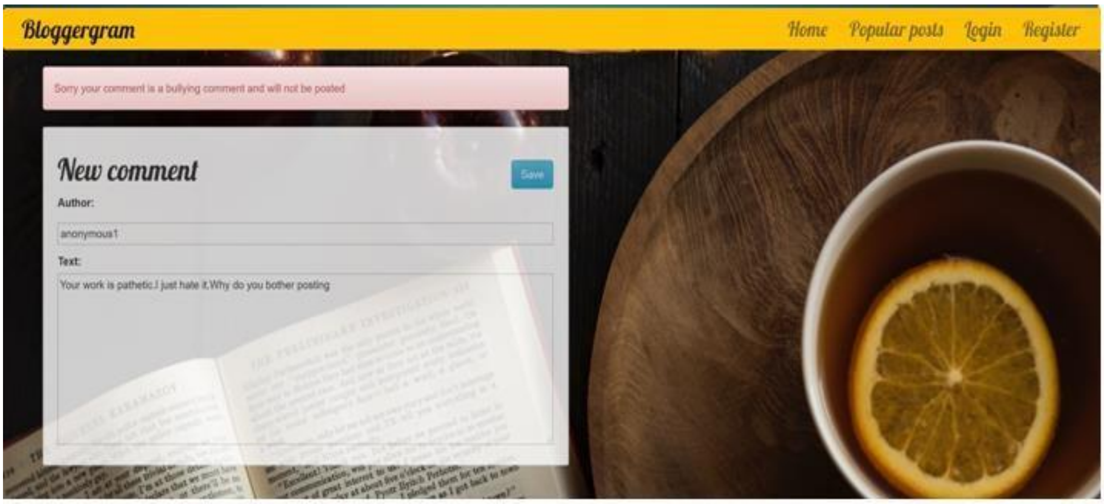


```
In [39]: get_prediction("I thank You.")

         Ouput of the model: [[0.10696062]]
         Predicted sentiment: nonbully.
         [[0.10696062]]

In [40]: get_prediction("You sing nicely.")

         Ouput of the model: [[0.00078872]]
         Predicted sentiment: nonbully.
         [[0.00078872]]

In [41]: get_prediction("You are an idiot.")

         Ouput of the model: [[1.]]
         Predicted sentiment: bully.

In [46]: get_prediction("You are crazy.")

         Ouput of the model: [[0.9998765]]
         Predicted sentiment: bully.

In [48]: get_prediction("You are very good.")

         Ouput of the model: [[0.34424666]]
         Predicted sentiment: nonbully.
         [[0.34424666]]
```

Fig 6.2 Training and Testing the model

After pre-processing the dataset, split the dataset into ratios for train and test. Accuracy is the performance measure to evaluate the classifier. So we are testing and training datasets to find the accuracy.

Bullying comments post



Sorry your comment is a bullying comment and will not be posted

Author:

anonymous1

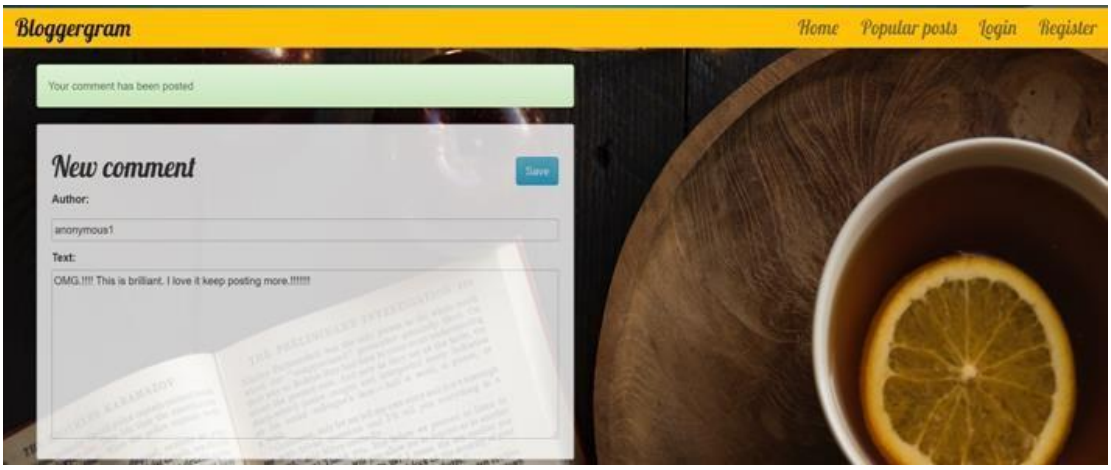
Text:

Your work is pathetic.I just hate it.Why do you bother posting

Fig 6.3 Bullying comments post

Whenever user tries to post a comment on social media like twitter, Instagram etc. which is a bully comments , then that comments will get delete and user will not be able to post that comment.

Non bullying comments post



Your comment has been posted

Author:

anonymous1

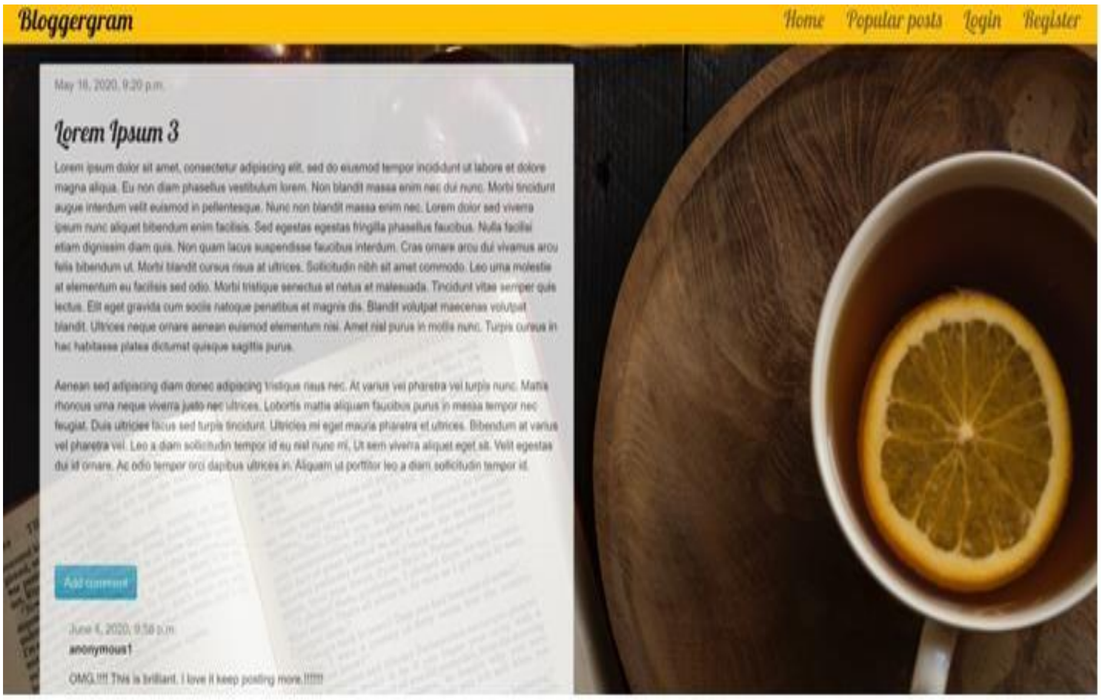
Text:

OMG.!!!! This is brilliant. I love it keep posting more.!!!!!!

Fig 6.4 Non bullying comments post

If an user tries to post a comment on social media like twitter, Instagram etc. which is not a bully comments, then that comment will be post successfully as those comments are non-bully. User can see also notification as “your comment has been posted “.

Non bullying comments is posted



May 16, 2020, 9:20 p.m.

Lorem Ipsum 3

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Eu non diam phasellus vestibulum lorem. Non blandit massa enim nec dui nunc. Morbi tincidunt augue interdum velit euismod in pellentesque. Nunc non blandit massa enim nec. Lorem dolor sed viverra ipsum nunc aliquet bibendum enim facilisis. Sed egestas egestas fringilla phasellus faucibus. Nulla facilisi etiam dignissim diam quis. Non quam lacus suspendisse faucibus interdum. Cras ornare arcu dui vivamus arcu felis bibendum ut. Morbi blandit cursus risus at ultrices. Sollicitudin nibh sit amet commodo. Leo urna molestie at elementum eu facilisis sed odio. Morbi tristique senectus et netus et malesuada. Tincidunt vitae semper quis lectus. Elit eget gravida cum sociis natoque penatibus et magnis dis. Blandit volutpat maecenas volutpat blandit. Ultrices neque ornare aenean euismod elementum nisi. Amet nisl purus in mollis nunc. Turpis cursus in hac habitasse platea dictumst quisque sagittis purus.

Aenean sed adipiscing diam donec adipiscing tristique risus nec. At varius vel pharetra vel turpis nunc. Mattis rhoncus urna neque viverra justo nec ultrices. Lobortis mattis aliquam faucibus purus in massa tempor nec feugiat. Duis ultricies lacus sed turpis tincidunt. Ultricies mi eget mauris pharetra et ultrices. Bibendum at varius vel pharetra vel. Leo a diam sollicitudin tempor id eu nisl nunc mi. Ut sem viverra aliquet eget sit. Velit egestas dui id ornare. Ac odio tempor orci dapibus ultrices in. Aliquam ut porttitor leo a diam sollicitudin tempor id.

June 4, 2020, 9:58 p.m.

anonymous1

OMG.!!!! This is brilliant. I love it keep posting more.!!!!!!

Fig 6.5 Non bullying comments is posted

After the non-bully comment posted , user can see that comment on that post.

6.1 Discussion

We proposed an approach to detect cyberbullying using machine learning techniques. We evaluated our model on convolution Neural Network.

- CNNs are usually known for their performance in computer vision. But it performs significantly well for text classification.
- One might argue about how RNN would be a better option. But RNNs are harder to train and RNN would be better for text generation, translation etc.
- CNNs on the other hand are much easier and CNN’s are good at extracting local and position-invariant features.
- CNN model is sufficient and better in terms of computation.

CHAPTER 7

TESTING

• Software testing defines as an activity to check whether the actual results match the expected results and to ensure that the system is defect free.

• The dataset was split and shuffled in such a way so as to include data from both cyberbullying and non-cyberbullying categories while testing the CNN model.

• Testing was done to check if the comments are being classified correctly, the time taken to classify and also the accuracy of the prediction model. (white box testing).

• Unit Testing was done one very single module of the application to check the working of all the functionalities.

7.1 Functional vs. Non-functional Testing

The goal of utilizing numerous testing methodologies in your development process is to make sure your software can successfully operate in multiple environments and across different platforms. These can typically be broken down between functional and non-functional testing.

Functional testing involves testing the application against the business requirements. It incorporates all test types designed to guarantee each part of a piece of software behaves as expected by using uses cases provided by the design team or business analyst.

Unit testing is the first level of testing and is often performed by the developers themselves. It is the process of ensuring individual components of a piece of software at the code level are functional and work as they were designed to. Developers in a test- driven environment will typically write and run the tests prior to the software or feature being passed over to the test team.

The functionalities of the application were checked in this testing method. Since the application is a simple blog application all the below functionalities were checked.

1. User Register and Login

For the user register and login, the password criteria the basic check needed to be done was to make sure an alert is generated if the fields of both the forms were left empty. Password field has to have a minimum of 6 characters and also if a valid email address has been entered

2. Creating and Uploading a post

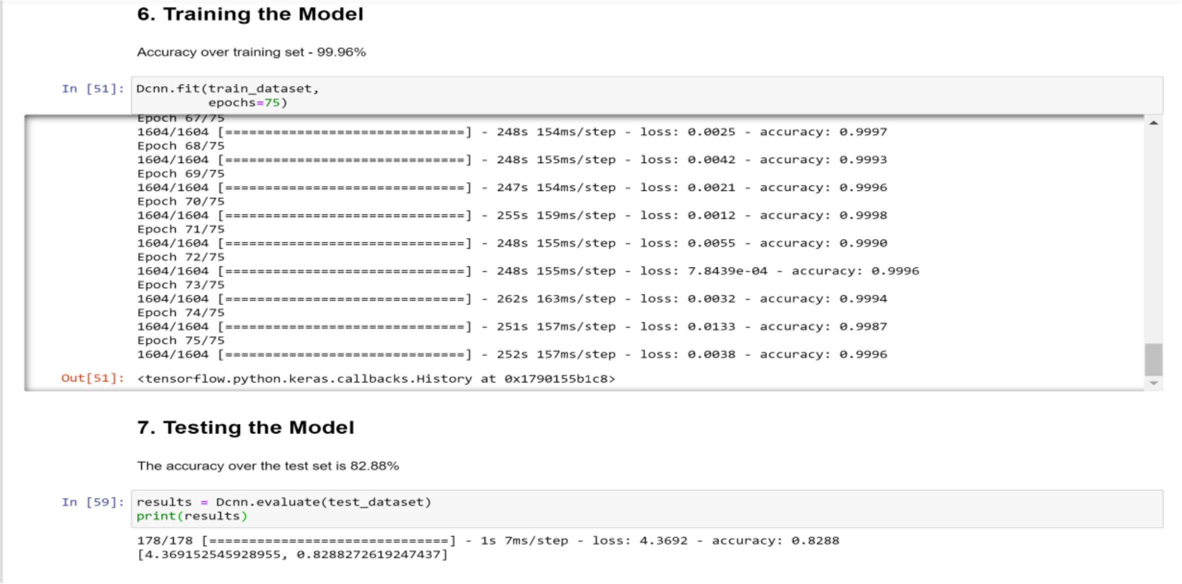
   Checks were made if the posts were correctly uploaded in the database and also if a post contains less than 50 characters if an error is thrown

3. Comments on post

   The comments on each post we checked if the predictive model is predicting correctly

   Apart from the application perspective of functional testing. The prediction model was tested on 10,000 data to check if it predicted correctly with the help of which the accuracy was determined.

   The prediction model was also tested separately i.e manually to check if it differentiates from a bully comment , a positive comment and also a comment being critical about the post .For example in the Fig7.3 the comment "Im confused did not understand the context of the word kill in sentence 2" is a critical comment and is not a bully comment even though the word kill has been mentioned.



6. Training the Model

Accuracy over training set - 99.96%

```
In [51]: Dcnn.fit(train_dataset,
                   epochs=75)
```

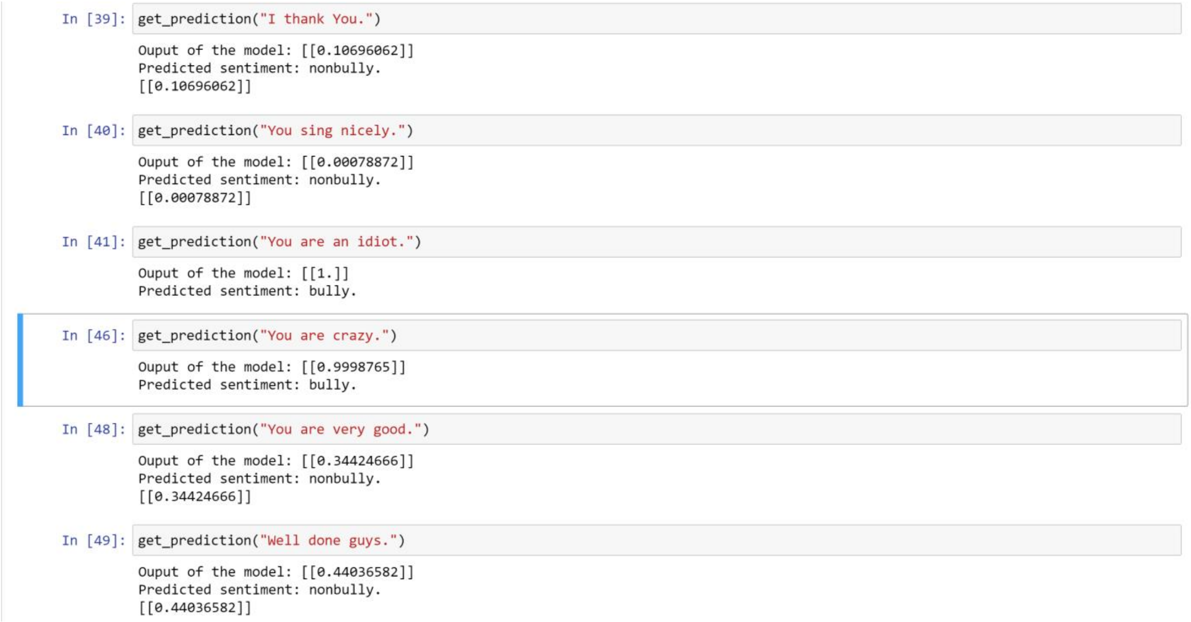
```
Epoch 67/75
1604/1604 [==============================] - 248s 154ms/step - loss: 0.0025 - accuracy: 0.9997
Epoch 68/75
1604/1604 [==============================] - 248s 155ms/step - loss: 0.0042 - accuracy: 0.9993
Epoch 69/75
1604/1604 [==============================] - 247s 154ms/step - loss: 0.0021 - accuracy: 0.9996
Epoch 70/75
1604/1604 [==============================] - 255s 159ms/step - loss: 0.0012 - accuracy: 0.9998
Epoch 71/75
1604/1604 [==============================] - 248s 155ms/step - loss: 0.0055 - accuracy: 0.9990
Epoch 72/75
1604/1604 [==============================] - 248s 155ms/step - loss: 7.8439e-04 - accuracy: 0.9996
Epoch 73/75
1604/1604 [==============================] - 262s 163ms/step - loss: 0.0032 - accuracy: 0.9994
Epoch 74/75
1604/1604 [==============================] - 251s 157ms/step - loss: 0.0133 - accuracy: 0.9987
Epoch 75/75
1604/1604 [==============================] - 252s 157ms/step - loss: 0.0038 - accuracy: 0.9996

Out[51]: <tensorflow.python.keras.callbacks.History at 0x1790155b1c8>
```

7. Testing the Model

The accuracy over the test set is 82.88%

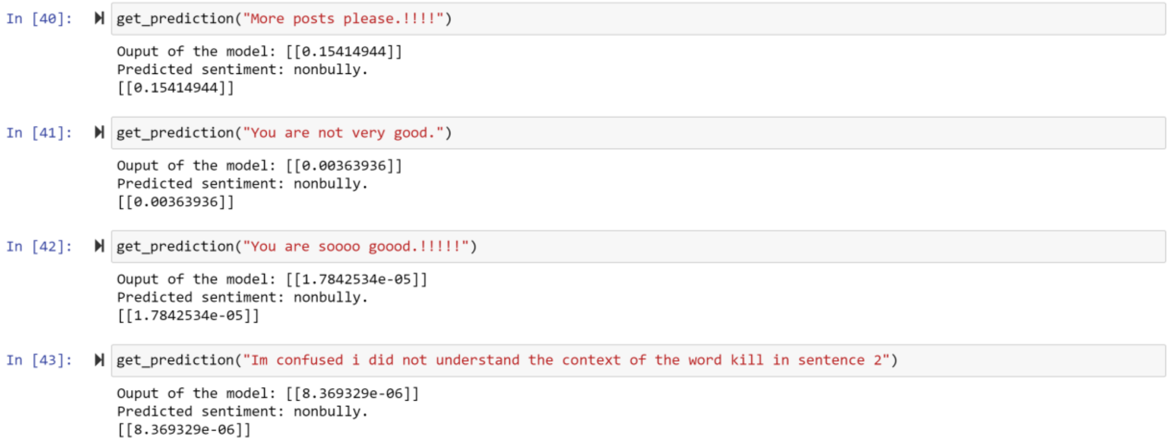
```
In [59]: results = Dcnn.evaluate(test_dataset)
         print(results)

178/178 [==============================] - 1s 7ms/step - loss: 4.3692 - accuracy: 0.8288
[4.369152545928955, 0.8288272619247437]
```

Fig 7.1: Testing the model



```
In [39]: get_prediction("I thank You.")

         Ouput of the model: [[0.10696062]]
         Predicted sentiment: nonbully.
         [[0.10696062]]

In [40]: get_prediction("You sing nicely.")

         Ouput of the model: [[0.00078872]]
         Predicted sentiment: nonbully.
         [[0.00078872]]

In [41]: get_prediction("You are an idiot.")

         Ouput of the model: [[1.]]
         Predicted sentiment: bully.

In [46]: get_prediction("You are crazy.")

         Ouput of the model: [[0.9998765]]
         Predicted sentiment: bully.

In [48]: get_prediction("You are very good.")

         Ouput of the model: [[0.34424666]]
         Predicted sentiment: nonbully.
         [[0.34424666]]

In [49]: get_prediction("Well done guys.")

         Ouput of the model: [[0.44036582]]
         Predicted sentiment: nonbully.
         [[0.44036582]]
```

Fig 7.2: First Phase of Testing



```
In [40]:   get_prediction("More posts please.!!!!")

           Ouput of the model: [[0.15414944]]
           Predicted sentiment: nonbully.
           [[0.15414944]]

In [41]:   get_prediction("You are not very good.")

           Ouput of the model: [[0.00363936]]
           Predicted sentiment: nonbully.
           [[0.00363936]]

In [42]:   get_prediction("You are soooo goood.!!!!!")

           Ouput of the model: [[1.7842534e-05]]
           Predicted sentiment: nonbully.
           [[1.7842534e-05]]

In [43]:   get_prediction("Im confused i did not understand the context of the word kill in sentence 2")

           Ouput of the model: [[8.369329e-06]]
           Predicted sentiment: nonbully.
           [[8.369329e-06]]
```

Fig 7.3: Second Phase of Testing

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

8.1 Conclusion

We have come to the conclusion that CNN is the most appropriate algorithm for our application due to the following advantages:

•CNN uses special convolution and pooling operations and performs parameter sharing.

•Enables CNN models to run on any device, making them universally attractive.

•CNN is very flexible and best for classification.

8.2 Future Scope

The project gives barrier-free access to the literature of research. It increases convenience, reach and retrieval power. This puts rich and poor on an equal footing for these key resources. Some of the features we like to include as a further step to our project are:

□ Images are also used nowadays for bullying.

□ It can be in the form of memes or any kind of morphed images.

□ A similar prediction model to classify such images can be done in the future

REFERENCES

Papers include:

[1] Kelly Reynolds - Using Machine Learning to Detect Cyberbullying ,Spring2012

[2] SemiuSalawu,YulanHe,andJoannaLumsden-Approachestoautomateddetection of cyberbullying-A survey,Proc of IEEE,October 2017.

[3] B.Sri Nandhini, J.I.Sheeba - Online Social Network Bullying Detection Using IntelligenceTechniques,ICACTA,2015

[4] Cynthia Van Hee ,Gilles Jacobs, Chris Emmery, Bart Desmet, Els Lefever, Ben Verhoeven, Guy De Pauw, Walter Daelemans, VeÂronique Hoste - Automatic detection of cyberbullying in social media text,PLoS ONE, October2018